An improved memory prediction strategy for dynamic multiobjective optimization

Jinhua Zheng
School of Computer Science
University of Xiangtan
Xiangtan, China
e-mail: jhzheng@xtu.edu.cn

Tian Chen
School of Computer Science
University of Xiangtan
Xiangtan, China
e-mail:1174739397@qq.com

Huipeng Xie
School of Computer Science
University of Xiangtan
Xiangtan, China
e-mail:126469584@qq.com

Shengxiang Yang
School of ComputerScience and Informatic
De Montfort University
Leicester, U.K.
e-mail: syang@dmu.ac.uk

***Abstract*—In evolutionary dynamic multiobjective optimization (EDMO), the memory strategy and prediction method are considered as effective and efficient methods. To handling dynamic multiobjective problems (DMOPs), this paper studies the behavior of environment change and tries to make use of the historical information appropriately. And then, this paper proposes an improved memory prediction model that uses the memory strategy to provide valuable information to the prediction model to predict the POS of the new environment more accurately. This memory prediction model is incorporated into a multiobjective evolutionary algorithm based on decomposition (MOEA/D). In particular, the resultant algorithm (MOEA/D-MP) adopts a sensor-based method to detect the environment change and find a similar one in history to reuse the information of it in the prediction process. The proposed algorithm is compared with several state-of-the-art dynamic multiobjective evolutionary algorithms (DMOEA) on six typical benchmark problems with different dynamic characteristics. Experimental results demonstrate that the proposed algorithm can effectively tackle DMOPs.**

Keywords- dynamic multiobjective optimization; memory strategy; prediction model;

I. INTRODUCTION

Many real-world problems are dynamic multiobjective optimization problems (DMOPs), with not only the conflict among multiple objectives but also the objective, constraint and related parameters may change over time [1], as well as the decision variables. As a consequence, the Pareto-Optimal Solutions (POS) and/or Pareto-Optimal Front (POF) may vary over time. A minimization problem is considered here without loss of generality. The dynamic multiobjective optimization problem [2] can be described as:

$$\begin{cases} \min F(x,t) = (f_1(x,t), f(x,t), ..., f_m(x,t))^T \\ s.t.\ \ g_i(x,t) \le 0\ i = 1,2,...p; h_j(x,t) = 0\ j = 1,2,...,q \\ subject\ to\ x \in \Omega \end{cases} \quad (1)$$

where m is the number of objectives, t = 0,1,2... represents discrete time instants, x is the decision variable vector, and Ω is decision space. $F(x,t)$ is the objective vector and consists of m time-varying objective functions that change intermittently. The function of $g_i \le 0$ and $h_j = 0$ present the set of inequality and equality constraints.

DMOPs have increasingly caused the attention of the research community in recent years. Multiobjective optimization evolutionary algorithms (MOEAs) have been widely used to solve DMOPs [3]-[6]. However, the changes in the POF and/or POS in DMOPs still pose significant challenges to traditional MOEAs. In a dynamic environment, traditional evolutionary algorithm makes the population gradually lose ability to adapt to environmental changes, the reason for this is that the purpose of traditional evolutionary algorithm is to make the population gradually converge to get a satisfactory solution set, but this would make the population lose diversity, especially in the later stages of the evolution [7], which are the challenges of traditional evolutionary algorithm. The difficulty for a multi-objective evolutionary algorithm (MOEA) solving DMOPs is that the algorithm may not re-locate the varied POS and/or POF before the environment changing again[8].How to track the Pareto optimal solution set after the change has always been an important and challenging issue. Dynamic MOEAs(DMOEAs) were further proposed to track a moving POF/POS quickly and obtain PSs that are uniformly distributed over time.

II. RELATED WORKS

Most of the existing DMOEAs are composed of combining classical MOEAs and effective dynamic handling techniques, including prediction-based [9]-[11], memory-based [12], [13], and diversity-based methods [14]-[16]. These strategies have shown competitive performances for solving DMOPs. But each is usually limited to solving a specific group of DMOPs. In many real-world DMOPs, the objective functions of problems change according to some regularity, rather than randomly between two consecutive environments [17], for this reason, the prediction-based and memory-based strategy is suitable for addressing DMOPs with predictable changes. If the dynamic changes of DMOPs are predictable, prediction-based strategies can learn the change patterns from the past changes to predict the new locations of the optimal solutions. Thus, the optimal solutions can quickly converge to the new PS.

Generally, the prediction-based strategy predicts the new locations of POS after detecting the environmental change, according to the previous values of POS. Recent years, scholars have proposed several prediction models, e.g., the autoregressive (AR) model [9], the Kalman Filter [10], the first-order difference model [2], and the predictive gradient strategy [12]. Memory-based approaches normally store the historical optimal information over the run and reuse the information subsequently when the new optima are sufficiently close to the historical ones [18]. Storing some optimal solutions or center points to reuse them in the new environment is the conventional method in memory-based strategies. As for the former method, it is difficult to determine the number of the individual and the way to choose them. And the reevaluation in the new environment costs additional computing resources. As for the latter method, if there is a similar environment, but actually the center point of it is far away from the idea one, it may cause a big prediction error.

Inspired by the memory strategy and prediction strategy, in this paper, an improved memory prediction strategy is proposed to tackle DMOPs. In this strategy, the moving direction of the previous two center points of an environment (the time variable of the environment keep constant in a time window) is considered as valuable information of the environment. It will be stored in an archive and environment detection will be conducted to detect environment change and identify similar environment. If there is a similar environment, the information will be reused to predict the new POS of the new environment.

The remainder of this paper is organized as follows. Section 3 presents the proposed improved memory prediction strategy. Experiments based on a set of benchmark functions and a comparative study is given in Section 4. Finally, the conclusion is given in Section 5.

III. IMPROVED MEMORY PREDICTION STRATEGY

A. Framework of The Proposed Algorithm

The overall framework of the proposed MOEA/D-MP is presented in Algorithm 1. To detect the environment change and identify the similarity between environments at different times, a Sensor-based detection method [19] is used in the algorithm. Besides, two archives are used to store historical information: *DC* stores the moving distance (a vector) of the population centers in the adjacent time, and *FV* stores the mean fitness value of all sensors (some individuals with constant decision variables) in different environments. When a change is detected, the mean value of sensors in the new environment will compare with the data stored in *FV* to find whether there is a similar environment in past times and *IND* store the index of similar environment. In dynamic reaction, a memory-prediction strategy and a prediction strategy are adopted to respond to similar and dissimilar environment changes, respectively. If no change occurs, the MOEA/D-DE is applied to optimize the static MOP with a fixed time variable.

Algorithm 1 Framework of MOEA/D-MP

Input: N (Population size)
Output: a series of approximated POFs
1: Set $t = 0$, $FV = \emptyset$, $DC = \emptyset$, $IND = -1$;
2:Initilaize a population P_t;
3:**while** stopping criterion is not met **do**
4: **if** change detected then
5: Compute the mean value of sensors: FV_{t+1}
6: Search in FV to find whether there is a FV_i similar to F_{t+1}, if it is, $IND = i$;
7: Update FV and DC (Section II-C);
7: Prediction process (Algorithm 2);
8: t = t + 1;
9: **else**
10: optimize the static MOP with MOEA\D-DE;
11: **end if**

B. Change Detection

Change detection method is a tool to detect the environment change and estimate the severity of a change. It enables evolutionary algorithms to respond to the environment change by taking the necessary steps to maintain their performance. In this paper, a Sensor–based detection, where a fix number of detectors is generated by Latin hypercube sampling [20] and remain the same in evolution process, is adopted to detect the change and identify similar change (line 4 of Algorithm 1). The fitness values of the fixed detector are re-evaluated in each generation and compared with the previously stored values. If the difference between them is greater than the set threshold, it means that an environment change has occurred. And then, the mean value of the sensors in the new environment, denoted as $FV_{t+1} = (f_1 , f_1 , . . . , f_1)^T$, will compare with the data in *FV*. If there is a F_i that satisfies (2), it can be considered that the environment corresponding to F_i is similar to the current environment (line 5 to 6 of Algorithm 1). And then, *IND* is set to *i*. Otherwise, *IND* keeps the initial value -1.

$$| FV^k_{t+1} — FV^k_i| < \varepsilon \quad \forall\, k = 1,2,\ldots,m \qquad (2)$$

where m is the objective number, ε is a threshold. Equation (2) means that if the difference between each dimension of the average objective values of the sensors in the two environments is really small, the two environments are considered to be similar.

C. Memory strategy

When handling DMOPs with periodic changes, algorithms can store information about historical optima like nondominated solutions or population centers to reuse them in the new environment. The memory-based strategy is widely used in DMOEAs, however, there are many difficulties to make good use of historical information. The optimal solutions stored in the memory may become outdated in the new environment and the reevaluation of these solutions cost additional computing resource. To properly make use of the historical information, in this paper a method (as described in Section III-B) is adopted to identify the similar environments. Besides, the moving distance of the population centers in the adjacent time is stored and will be reused when there is a similar environment.

When an environmental change is detected, the information of the last old environment needs to be stored in the archive. The mean objective value of sensors F_t is added to *FV*, and the corresponding moving distance of population center DC_t is added to *DC*. DC_t is defined by the following formula:

$$DC_t = C_t — C_{t-1} \tag{3}$$

$$C_t^i = \frac{1}{|A_t|}\sum_{x_t \in A_t} x_t^i \tag{4}$$

where C_t, which is a vector, represents the population center of the last generation at time t and A_t is a set of nondominated solutions in the population. It should be noticed that from the above equations, the moving distance of population center DC_t is defined by the difference of population center at time t (current environment) and that at time t—1. Suppose there is an environment similar to the new environment, we should determine whether to use the new information to replace the old one. First of all, we should clearly realize the purpose of storing DC_t, which is to use historical information to guide the prediction process in the new environment. So we need a measure for reflecting the degree of DC_t in improving the prediction process. For this purpose, $\delta(t)$ is designed, which is indicated by the degree of the difference in the objective values of population after the use of DC_t and the population after optimization of the static algorithm. $\delta(t)$ is presented in the following:

$$\delta(t) = \max\left(\frac{|\bar{F}_{Be}^j - \bar{F}_{Af}^j|}{|\bar{F}_{Be}^j|}\right) \quad j = 1, 2, \ldots, m \tag{5}$$

$$\bar{F}_k = \frac{1}{N}\sum_{x_i \in P_k} F(x_i) \tag{6}$$

where $\bar{F}_{Be}^j$ is the mean value of jth objective of the population after prediction, $\bar{F}_{Af}^j$ is the mean value of jth objective of the population after optimization of the static algorithm. N is the number of population and m is the number of objective. If $\delta(t)$ is larger than the predefined threshold λ, the effect of DC_t in the prediction process can be considered small, so the new one should replace it. When the size of the archive *DC* exceeds the predefined size, the first-in-first-out strategy is applied to update it, and the update of the archive *FV* corresponds to *DC*.

D. Prediction Process

When an environment change is detected, a good prediction model should be able to generate an approximation of the new location of POS in the new environment to speed up the convergence and maintain good diversity. In this paper, the proposed prediction model is an improved memory prediction model which exploits the historical information to predict the POS in the new environment. If there is no similar environment in *DC*, the moving direction of the previous two consecutive population centers is adopted to predict the new location of POS. To avoid the poor individuals mislead the prediction process, the center of nondominated solutions is seen as the position of the population center (as described in (4)). The predicted location of the individuals is generated as follows:

$$x_{t+1}^i = x_t^i + C_t^i - C_{t-1}^i + N(0, \sigma_t) \tag{7}$$

where $i = 1,2,..., n$ is the index of the decision variable and n represents the dimension of decision variable. $N(0,\sigma_t)$ is a Gaussian noise to increase the probability of the predicted solution to locate in the POS. σ_t is defined by:

$$\sigma_t = \left(\sum_{i=1}^{n} |D_t^i|^2\right)^{1/2} \tag{8}$$

If there is an environment in *DC* similar to the new environment, the information of it can be reused to help the prediction of the initial population of the new environment. First, a search is conducted in *FV* to find whether there is a FV_i similar to FV_{t+1} (FV_i *and* FV_{t+1} satisfy (2)), which means the ith environment is similar to the new environment. The similar environment in *DC* is indicated by *IND*. Second, the moving distance of the similar environment, denoted by DC_{IND}, is reused in the prediction process of the new environment. Then the value of the decision variables of the individuals at time t+1 can be generated by the following formula:

$$x_{t+1}^i = x_t^i + D_{IND}^i \tag{9}$$

The overall prediction procedure is presented in Algorithm 2. If *IND* (the initial value of *IND* is -1) is larger than 0, which means there is a similar environment, the prediction process of (9) is conducted. Otherwise, (7) is conducted to generate individuals based on the moving direction of previous two center points.

The overall prediction procedure is presented in Algorithm 2. If *IND* (the initial value of *IND* is -1) is larger than 0, which means there is a similar environment, the prediction process of (9) is conducted to reuse the moving direction of the previous two center points of the similar environment (stored in archive *DC*). Otherwise, (7) is conducted to generate individuals based on the moving direction of the previous two center points. To maintain good diversity of population and alleviate the effect of prediction errors, 20% of individuals of the population are randomly generated.

Algorithm 2 Pseudocode of prediction procedure

Input: *IND* (index of similar environment), P_t (population at t),

Output: P_{t+1} (initial population of time t+1)

1:**If** *IND* < 0

2: Generate 0.8N individuals by (7);

3:**else**

4: Generate 0.8N individuals by (9);

5: **end if**

5: Generate 0.2N individuals by randomly sampling from the decision space

8:Construct the population P_{t+1} by the N generated individuals;

10: *IND* = -1;

11: Return $P_{t+1;}$

IV. Experimental Study

A. Test Probleams and Performance Metric

Five test problems including FDA1, FDA3 [21], dMOP1 [22], JY1, JY3 and JY7 [23] are used to assess our proposed algorithm in comparison with other algorithms. The time instance t involved in these problems is defined as $t=(1/n_t)\lfloor \tau/\tau_t \rfloor$,where n_t , τ and τ_t represent the severity of change, the frequency of change, and the iteration counter, respectively.

In our experimental studies, we adopt the performance metric called Inverted Generational Distance (IGD)[24], as they can help deeply investigate algorithms performance regarding convergence, distribution, and diversity measures both the convergence and diversity of found solutions by an algorithm. Let POF be a set of uniformly distributed points in the true POF, and POF* be an approximation of the POF. The IGD is calculated as follows:

$$IGD = \frac{1}{n_{POF}} \sum_{i=1}^{n_{POF}} d_i \qquad (10)$$

where n_{POF} = |POF|, d_i is the Euclidean distance between the ith member in POF and its nearest member in POF*.

B. Compared Algorithm and Parameter Settings

Three popular DMOEAs are used for comparison in our empirical studies. They are the MOEA based on decomposition (MOEA/D-DE) [25], MOEA/D-DE with Kalman Filter prediction (MOEA/D-KF) , and population prediction strategy (PPS), representing different classes of metaheuristics

The parameters of the MOEAs considered in the experiment were referenced from their original papers. The general parameter settings are shown in Table I. Some key parameters in these algorithms were set as follows.

TABLE I. PARAMETER SETTINGS

Parameters	**Values**
Population Size (N)	100
Dimension of decision variables (D)	10
Environment change change severity (n_t)	10
Distribution index in mutation (η_t) & mutation rate (P_m)	20, 1/n
Scaling Factor (F) & crossover rate (C_r)	0.5, 1.0
Number of time windows and independent running times	100, 10

- The parameters for MOEA/D-DE are implemented as guided in [22], when a change is detected the randomly generated population is as the initial population.
- In PPS, 10 individuals are randomly selected from the existing population as the detectors.
- MOEA/D-KF adopts the fixed detector approach, where 10 randomly generated individuals in the decision space are employed to detect the changes.
- As for MOEA/D-MP, the threshold ε in (2) is set to 10^{-4}, and
- In the experiment, each algorithm was run 10 times independently on each test problem. The total number of generations was set to 100τ_t, which ensured 100 environmental changes in each run.

C. Comparative Study

Table II shows the obtained average IGD values and standard deviations over 10 runs by four algorithms on the six test instances, where the best values are highlighted in bold. The Wilcoxon rank-sum test [26] was conducted to point out significance between different results at the 0.05 significance level. It is obvious that MOEA/D-MP performed best on the majority of the six instances, implying that it has the best tracking ability of changing POS and/or POF in most cases. The POS and POF of FDA3 changed over time, in which environmental changes shifted the POS and affected the density of points on the POF. The randomly reinitialized approach for MOEA/D-DE was better than prediction when the change frequency was relatively slow (τt=20) on this problem. However, when the change frequency was fast, the prediction approach could enhance the searching efficiency, in which the moving direction and the moving step-size may be predicted correctly. As shown in the table MOEAD-MP performs best in FDA3 with τ_t=25 and PPS performs best in FDA3 with τ_t=30. For JY3, that is a problem with time-varying non-monotonic dependencies between any two decision variables, the four algorithms perform worse in it. Therefore, a good method is needed for the four algorithms to address these kinds of problems, which is one of our future goals. As for JY7, MOEAD-MP performs slightly worse than MOEA/D-KF, and better than the others. The reason may be that the multimodality of JY7 may make the algorithms fall into the local optima, and form the table we can see that the standard deviations of the four algorithms on JY7 are relatively big.

TABLE II. STATISTICAL RESULT OF IGD METRIC FOR THREE STRATEGIES

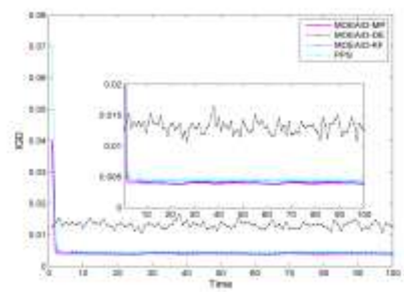
Prob.	τ_t	PPS	MOEA/D	MOEA/D-KF	MOEAD-MP
FDA1	20	1.885e-2 (1.434e-2)‡	3.279e-2 (2.010e-3)‡	5.709e-3 (6.494e-5)‡	**4.305e-3 (1.707e-4)** ‡
	25	8.787e-3 (4.170e-3)‡	1.879e-2 (1.528e-3)‡	5.143e-3 (4.883e-5)‡	**4.165e-3 (2.621e-4)**
	30	6.324e-3 (2.319e-3)‡	1.256e-2 (6.800e-4)‡	4.870e-3 (3.141e-5)‡	**3.956e-3 (2.261e-5)**
FDA3	20	3.314e-2 (2.682e-2)‡	**1.822e-2 (8.488e-4)**‡	1.932e-2 (1.259e-3)‡	1.906e-2 (4.672e-4)
	25	1.891e-2 (9.364e-3)‡	1.673e-2 (1.144e-3)‡	1.812e-2 (1.258e-3)‡	**1.513-2 (3.553e-4)**
	30	**8.031e-3 (1.017e-3)**‡	1.476e-2 (8.983e-4)‡	1.633e-2 (9.649e-4)‡	9.827e-3 (3.839e-4)
DMOP1	20	1.816e-2 (2.392e-2)‡	3.016e-2 (1.647e-3)‡	5.664e-3 (2.678e-4)†	**5.461e-3 (7.361e-4)**
	25	5.472e-3 (6.603e-4)‡	1.845e-2 (6.909e-4)‡	4.923e-3 (3.296e-4)†	**4.733e-3 (6.116e-4)**
	30	5.185e-3 (9.671e-4)†	1.310e-2 (1.792e-4)‡	4.443e-3 (1.461e-4)†	**4.287e-3 (5.634e-4)**
JY1	20	1.970e-2 (1.100e-2)‡	2.342e-2 (5.922e-4)‡	7.252e-3 (6.851e-5)‡	**6.309e-3 (3.824e-5)**
	25	1.403e-2 (8.469e-3)‡	1.473e-2 (3.265e-4)‡	6.718e-3 (6.287e-5)‡	**5.957e-3 (1.266e-5)**
	30	8.136e-3 (4.474e-3)‡	1.123e-2 (1.957e-4)‡	6.416e-3 (3.739e-5)‡	**5.771e-3 (5.308e-6)**
JY3	20	2.525e+ (2.273e-2)‡	3.352e-1 (5.423e-3)†	**3.082e-1 (2.563e-3)**	3.099e-1 (3.269e-3)
	25	2.505e+ (2.123e-3)‡	3.288e-1 (5.695e-3)†	3.100e-1 (1.465e-3)†	**3.064e-1 (3.402e-3)**
	30	2.503e+ (3.769e-3)‡	3.229e-1 (5.847e-3)†	3.098e-1 (1.604e-3)†	**3.083e-1 (3.081e-3)**
JY7	20	1.174e+1 (3.219e-1)‡	8.448e-1 (3.320e-2)‡	**3.834e-1 (3.497e-1)**	5.651e-1 (3.501e-1)
	25	1.146e+1 (2.467e-1)‡	8.350e-1 (5.006e-2)‡	**1.776e-1 (1.996e-1)**	5.167e-1 (2.654e-1)
	30	1.092e+1 (2.059e-1)‡	8.055e-1 (5.437e-2)‡	**3.442e-1 (2.887e-1)**	4.744e-1 (3.522e-1)

‡ and † indicate MOEAD-MP performs significantly better than and equivalently to the corresponding algorithm, respectively.

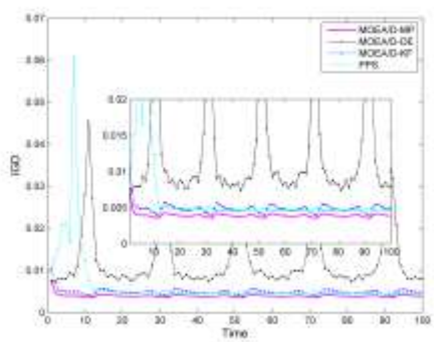
Apart from the tabular information, the tracking of the IGD values with the environmental change obtained by four algorithms for JY1 JY3 with n_t=10; τ_t =30 are shown in Fig. 1. It can be seen that, MOEAD-MP is stable in responding to the environment in most problems.

V. CONVLUSIONS

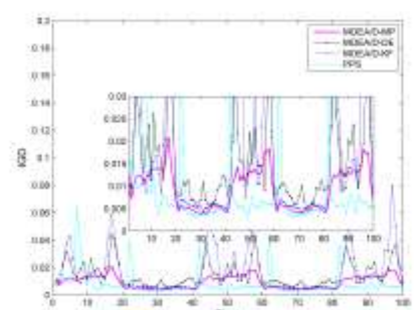
In this paper, an improved memory prediction strategy was proposed to solve dynamic multiobjective optimization problems. The memory prediction model was incorporated into MOEA/D algorithm, in which the model was used to predict the new location of the POS based on the historical information. The change detection is used to detect environment change and identify similar environment. The moving direction of the previous two center points of the similar environment is reused to predict the new POS in the new environment. If there is no similar environment a simple center point prediction is adopted. And a portion of population individuals are randomly reinitialized to enhance the diversity. The experimental results showed that this proposed model has strongly competitive power to solve the majority DMOPs. The proposed prediction strategy illustrates that a proper prediction approach is effective to enhance the tracking ability of MOEAs in dynamic environments.



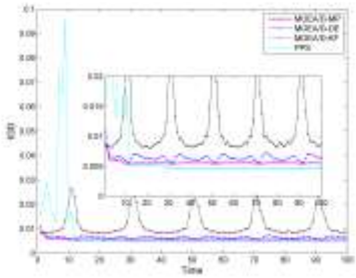
a) FDA1



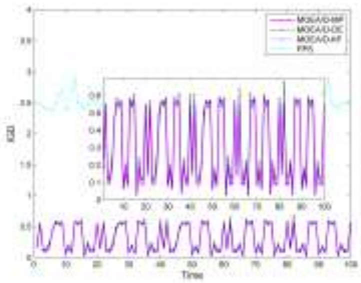
b) FDA3



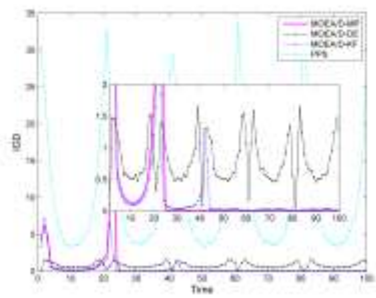
c) DMOP1



d) JY1



e) JY3



f) JY7

Figure 1. Evolutionary curves of average IGD values for six problems.

REFERENCES

[1] Yaochu Jin and Jurgen Branke. 2005. Evolutionary optimization in uncertain environments-a survey. IEEE Transactions on Evolutionary Computation 9, 3: 303–317.

[2] M. Farina, K. Deb, and P. Amato, “Dynamic multiobjective optimization problems: test cases, approximations, and applications,” IEEE Transactions on Evolutionary Computation, 2004.8(5):425-442

[3] Leilei Cao, Lihong Xu, Erik D Goodman, and Hui Li. 2017. A First-Order Difference Model-Based Evolutionary Dynamic Multiobjective Optimization.Proceding of SEAL 2017, LNCS 10593, Springer International Publishing AG 2017, 644–655K.

[4] R. Shang, L. Jiao, Y. Ren, L. Li, L. Wang, Quantum immune clonal coevolutionary algorithm for dynamic multiobjective optimization, Soft Comput. 18(4) (2014) 743–756.

[5] J. Wei, Y. Wang, Hyper rectangle search based particle swarm algorithm for dynamic constrained multi-objective optimization problems, in: 2012 IEEE Congress on Evolutionary Computation, 2012, pp. 1–8

[6] A. Zhou, Y. Jin, Q. Zhang, A population prediction strategy for evolutionary dynamic multiobjective optimization, IEEE Trans. Cybern. 44 (1) (2014)40–53.

[7] T.T. Nguyen, S.X. Yang and J. Branke, “Evolutionary dynamic optimization: A survey of the state of the art,”Swarm and Evolutionary Computation, 2012.6:1-24.

[8] Shouyong Jiang and Shengxiang Yang. 2017. A Steady-state and Generational Evolutionary Algorithm for Dynamic Multiobjective Optimization. IEEE Transactions on Evolutionary Computation 21, 1: 65–82.

[9] I. Hatzakis, D. Wallace, Dynamic multi-objective optimization with evolutionary algorithms:a forward-looking approach, in: 2006 Conference on Genetic and Evolutionary Computation, 2006, pp. 1201–1208.

[10] A. Muruganantham, K.C. Tan, P. Vadakkepat, Evolutionary dynamic multiobjective optimization via kalman filter prediction., IEEE Trans. Cybern. 46(12) (2016) 2862–2873.

[11] A. Zhou, Y. Jin, Q. Zhang, B. Sendhoff, E. Tsang, Prediction-based population re-initialization for evolutionary dynamic multi-objective optimization, in:2007 International Conference on Evolutionary Multi-Criterion Optimization, 2007, pp. 832–846.

[12] W.T. Koo, K.G. Chi, K.C. Tan, A predictive gradient strategy for multiobjective evolutionary algorithms in a fast changing environment, Memetic Comput.2 (2) (2010) 87–110.

[13] S. Sahmoud, H.R. Topcuoglu, A memory-based NSGA-II algorithm for dynamic multi-objective optimization problems, in: 2016 European Conference on the Applications of Evolutionary Computation, 2016, pp. 296–310.

[14] C.R.B. Azevedo, A.F.R. Arajo, Generalized immigration schemes for dynamic evolutionary multiobjective optimization, in: 2011 IEEE Congress on Evolutionary Computation, 2011, pp. 2033–2040.

[15] R. Azzouz, S. Bechikh, and L. B. Said, “A dynamic multi-objective evolutionary algorithm using a change severity-based adaptive population

[16] management strategy,” Soft Comput, aug 2015Biswas, S. Das, S. Kundu, G.R. Patra, Utilizing time-linkage property in DOPs: an information sharing based artificial bee colony algorithm for tracking

[17] K. Deb, N. Rao, S. Karthik, Dynamic multi-objective optimization and decision-making using modified NSGA-II: a case study on hydro-thermal powerscheduling, in: 2007 International Conference on Evolutionary Multi-Criterion Optimization, 2007, pp. 803–817.

[18] J. Branke, Memory enhanced evolutionary algorithms for changing optimization problems, in: 1999 Congress on Evolutionary Computation, 1999,pp. 1875–1882.

[19] Richter, Hendrik . "Detecting change in dynamic fitness landscapes." Evolutionary Computation, 2009. CEC '09. IEEE Congress on IEEE, 2009.

[20] Mckay M D , Conover R J B J . A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code[J]. Technometrics, 1979, 21(2):239-245.

[21] S.B. Gee, K.C. Tan, H.A. Abbass, A benchmark test suite for dynamic evolutionary multiobjective optimization, IEEE Trans. Cybern. 47 (2) (2017) 461‐472.

[22] C. Goh, K.C. Tan, A competitive-cooperative coevolutionary paradigm for dynamic multiobjective optimization, IEEE Trans. Evol. Comput. 13 (1) (2009)103‐127.

[23] Shouyong Jiang and Shengxiang Yang. 2017. Evolutionary Dynamic Multiobjective Optimization: Benchmarks and Algorithm Comparisons. IEEE Transactions on Cybernetics 47, 99: 198‐211.

[24] Y. Yuan, H. Xu, B. Wang, B. Zhang, and X. Yao, “Balancing convergence and diversity in decomposition-based many-objective

[25] H. Li, Q. Zhang, Multiobjective optimization problems with complicated pareto sets, MOEA/D and NSGA-II, IEEE Trans. Evol. Comput. 13 (2) (2009)284‐302.

[26] Wilcoxon, Frank. "Individual comparisons by ranking methods." *Breakthroughs in statistics*. Springer, New York, NY, 1992. 196-202.